

A multi-lane TASEP model for crossing pedestrian traffic flows

H.J. Hilhorst and C. Appert-Rolland

Laboratoire de Physique Théorique, bâtiment 210
Université Paris-Sud and CNRS, 91405 Orsay Cedex, France

March 3, 2013

Abstract

A one-way *street* of width M is modeled as a set of M parallel one-dimensional TASEPs. The intersection of two perpendicular streets is a square lattice of size $M \times M$. We consider hard core particles entering each street with an injection probability α . On the intersection square the hard core exclusion creates a many-body problem of strongly interacting TASEPs and we study the collective dynamics that arises. We construct an efficient algorithm that allows for the simulation of streets of infinite length, which have sharply defined critical jamming points. The algorithm employs the ‘frozen shuffle update’, in which the randomly arriving particles have fully deterministic bulk dynamics. High precision simulations for street widths up to $M = 24$ show that when α increases, there occur jamming transitions at a sequence of M critical values $\alpha_M^M < \alpha_{M-1}^M < \dots < \alpha_1^M$. As M grows, the principal transition point α_M^M decreases roughly as $\sim (\log M)^{-1}$ in the range of M values studied. We show that a suitable order parameter is provided by a reflection coefficient associated with the particle current in each TASEP.

Keywords: exclusion process, crossing flows, pedestrian traffic, frozen shuffle update

1 Introduction

Pedestrian motion in dense environments is of both theoretical and practical interest. Instances of applications are shopping streets, waiting lines, crowds that enter or leave a confined space, and so on. Under such circumstances simplified models may help understand the behavior of individuals as well as the collective behavior that results from it [1, 2, 3, 4]. A particular class of such models is based on cellular automata [5, 6, 7, 8]. For modeling unidirectional *one-dimensional* traffic, whether it be of particles, vehicles, or pedestrians [9], one popular tool is the Totally Asymmetric Simple Exclusion Process (TASEP) [10]. This stochastic process belongs to the larger class of models of random walkers with hard core interactions and various different TASEP versions have been studied by physicists and mathematicians alike since several decades. Specifically, a TASEP is a system of hard core particles that advance along a linear lattice in a single direction. The TASEP may be used as a building block for traffic flow models in more complicated geometries. There is a rich literature on junctions and bifurcations [11, 12, 13, 14], as well as on intersections [15, 16, 17, 18, 19, 20] of TASEPs. In those approaches the interaction between the different TASEPs is ‘weak’ in the sense that they are coupled only on a set of generally well-separated sites. Strongly interacting TASEPs were considered, in particular, in the so-called BHL model proposed by Biham *et al.* [21], which in turn has given rise to an offspring of variants on some of which we will comment.

Although pedestrian traffic flow is our basic motivation, we present below a model which, because of its simplicity, has an intrinsic interest that extends beyond this particular application. From a wider perspective it is an example of a driven nonequilibrium system, and, contrary to many other such systems that have been studied in the literature and that have stochastic dynamics [22, 23], this one is deterministic.

We define a *street of width M* as a set of M parallel linear lattices to be called *lanes*, each of which carries a TASEP. We wish to study what happens when two such streets, both thought of as being of infinite length, intersect perpendicularly. Figure 1 shows the geometry for the finite length case. Particles in the horizontal street are injected from the left onto empty sites with a probability α per time step. Particles in the vertical street are injected from below with the same probability. The resulting average incoming current in a lane will be denoted by $J^{\text{free}}(\alpha)$.

The intersection area of the two streets is a finite $M \times M$ square lattice. On this square we have a problem of *strongly* interacting TASEPs, whose study is the purpose of this work. Lanes are numbered by an index $m = 1, 2, \dots, M$ from the outer ones inward. Unless symmetry were bro-

ken spontaneously, the m th horizontal and m th vertical lane are statistically identical. One basic question here, as in any other traffic flow model, is to determine the outgoing currents $J_m^M(\alpha)$ as a function of α when the system is in a stationary state. Contrary to the incoming current, the outflow must be expected to be m and M dependent. A lane is said to be in a *free flow phase* when its outgoing current is equal to the incoming current, that is, when $J_m^M(\alpha) = J^{\text{free}}(\alpha)$. At sufficiently low α free flow is to be anticipated for all m ; however, it is inevitable that when α increases above a certain threshold, part or all of the lanes undergo a jamming transition.

The update scheme is an essential part of the definition of any TASEP. We choose to employ here the frozen shuffle update [24, 25, 20], whose characteristic is that in each time step the particles are updated according to a sequence fixed once and for all. Particles entering the system are inserted in this sequence and particles leaving are deleted from it. This update, originally proposed as an alternative to parallel update¹, has the advantage of eliminating the algorithmic conflicts that arise when in the same time step two particles have the same target site. It conserves the advantage of the parallel update in that it bounds the fluctuations that random sequential update would cause. In our present implementation of the frozen update scheme all allowed moves are carried out with probability one and hence the bulk dynamics is that of a cellular automaton: it is fully deterministic.

Under frozen shuffle update, once a particle has left the intersection square it continues unimpeded at unit speed, and hence the street segments beyond the intersection square need not be considered. The street segments leading up to this square, however, play an essential role, since they are the place where waiting lines may develop, whether temporarily or permanently. Such waiting lines are an intrinsic part of the traffic flow problem and we wish to fully account for them. In any simulation the ingoing street segments are necessarily of finite length L , as shown in figure 1. A key observation is, however, that sharply defined jamming transition points on the α axis can exist only in the limit $L \rightarrow \infty$. Indeed, as previous work [20] has shown, there is a sharp transition point for crossing streets even of width $M = 1$, as long as their length L is infinite. This being so, the present work does the following.

First, by combining theoretical and algorithmic arguments, we show that it is possible to integrate out the degrees of freedom in the half-way infinite street segments leading up to the intersection square. There then results an efficient algorithm that simulates the particle motion on the *finite* $M \times M$ square, subject to boundaries on its left and lower edge that represent the

¹Frozen shuffle update may also be seen as a variant of *random* shuffle update [26, 27]; in the latter a new random particle order is drawn before each time step.

incoming street segments of infinite length, $L = \infty$. Finite size effects have thus been eliminated. The appropriate boundary conditions are derived and formulated in terms of *memory variables* to be defined in section 3.

Secondly, we present the results of the simulation of this interacting street model as a function of its two parameters α and M . We obtain the outgoing currents $J_m^M(\alpha)$ and find that when α increases from 0 to 1, there appear M critical values

$$\alpha_M^M < \alpha_{M-1}^M < \dots < \alpha_1^M \quad (1.1)$$

at which, successively, the lanes with indices $m = M, M-1, \dots, 1$ become jammed. We will refer to α_m^M as the *principal critical point* of the size M intersection. For α beyond the critical value α_m^M , the outflow of particles in the m th horizontal and vertical lane cannot keep up with the incoming current and in those lanes ever growing waiting lines develop.

We relate the incoming and outgoing current in lane m by

$$J_m^M = (1 - R_m^M) J^{\text{free}}. \quad (1.2)$$

The product $R_m^M J^{\text{free}}$ in (1.2) may be interpreted as the reflected current² in the m th lane and we therefore call R_m^M its *reflection coefficient*. We will show that there exists a simple theoretical relation between this coefficient and the memory variables occurring in the boundary condition. In our Monte Carlo work we deduce the critical point values α_m^M from the α dependence of R_m^M through the criterion

$$\begin{aligned} R_m^M(\alpha) &= 0, & \alpha &\leq \alpha_m^M, \\ R_m^M(\alpha) &> 0, & \alpha &> \alpha_m^M, \quad m = 1, 2, \dots, M. \end{aligned} \quad (1.3)$$

For $\alpha_m^M < \alpha < 1$ the coefficient $R_m^M(\alpha)$ increases monotonously and we may appropriately consider it as an order parameter. The main results of our simulations are the determination of the critical points α_m^M and of the curves $R_m^M(\alpha)$.

This paper is set up as follows. In section 2 we define the rules of motion for what we will call the ‘full’ algorithm, that is, the one executed on the lattice of figure 1, which has a finite L . In section 3 the memory variables are introduced; we then show how for $L = \infty$ the full algorithm gives rise to a ‘reduced’ algorithm, restricted to the $M \times M$ interaction square, and with boundary conditions formulated in terms of these new variables. In section 4 we establish the theoretical expressions, valid for each lane separately, that relate the memory variable to the transmitted current and the speed of propagation of the reflected current. In section 5 we present and discuss simulation results for lattices up to $M = 24$. In section 6 we make several final remarks and conclude.

²This even though all particles move only in a single direction; see section 4.3.

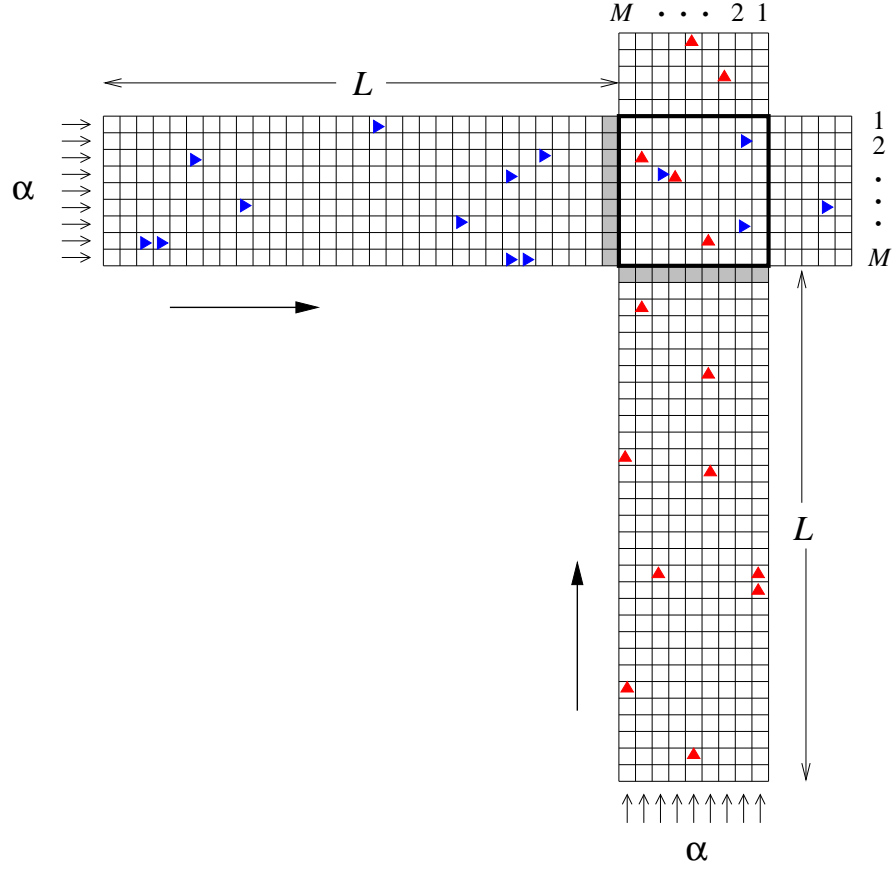


Figure 1: Intersection of two one-way streets of width M . The x particles, moving to right, and the y particles, moving upward, are represented by blue and red triangles, respectively. The control parameter α is the injection probability per time step for each lane. Small arrows point towards the ‘injection’ sites. Lanes are numbered $m = 1, 2, \dots, M$. The region bordered by the heavy solid line is the ‘intersection square’. The ‘entrance sites’ defined in section 3 are shaded.

2 Intersecting street model

We consider the lattice of figure 1 showing a ‘horizontal’ and a ‘vertical’ one-way street labeled ‘x’ and ‘y’, respectively, and each composed of M parallel lanes. A lane is identified by an index pair (m, δ) with $m = 1, 2, \dots, M$ and $\delta = ‘x’$ or ‘y’. The length L of the street segments leading up to the intersection is a large finite number³. The two streets intersect according to a square of $M \times M$ sites. Two types of hard-core particles move on the lattice, x particles arriving from the left along the horizontal street, and y particles arriving from below along the vertical street. New particles may be injected onto the sites of the leftmost column of the horizontal street and of the bottom row of the vertical street; we will refer to these as the *injection* sites. We let $a > 0$ denote the *injection rate* of particles onto an empty injection site. The ensuing *injection probability* $\alpha = 1 - e^{-a}$ is the probability that an injection site, when empty at a certain instant of time t , will get occupied during the unit time interval $(t, t + 1)$ that follows. We will employ α rather than a as the control parameter.

2.1 Full algorithm

The rules for the particle motion on this lattice are as follows. The algorithm is a succession of time steps labeled by an integer time step index $s = 1, 2, 3, \dots$. The s th time step takes the system from time $t = s - 1$ to $t = s$. It executes all events (particle moves, injections and deletions) that occur during the time interval $s - 1 < t < s$.

With the frozen shuffle update [24, 25, 20] employed here, either at the initial time $t = 0$, or else when it enters the system, the i th particle in lane (m, δ) is assigned, in a way to be described below, a random *phase* $\tau_i^{m\delta} \in (0, 1)$; it keeps this phase as a fixed attribute. The phases define in which order the particles are updated at each time step. For practical reasons the algorithm decouples each time step update into a ‘bulk’ step and an ‘injection’ step, both to be described now.

2.1.1 Bulk step

The s th bulk time step consists of a sweep through the ‘update sequence’, that is a list, common to all lanes, of all particles present in the system at the beginning of that time step, ordered according to increasing phases. Thus, during the bulk step, each particle present in the system is visited exactly once. This bulk step parallels the evolution of the continuous time variable t , the turn for particle i to be updated being identified with the instant of time $t = s - 1 + \tau_i^{m\delta}$.

³In section 3 we will set $L = \infty$.

General case. The case of a general particle i in a lane (m, δ) is simple. When at time $t = s - 1 + \tau_i^{m\delta}$ its turn to be updated has come, then if at that instant of time its target site is occupied, it does not move; if its target site is empty, it moves.

First special case. A particle i occupying the last site of its lane (an ‘exit site’) has no target site. When its turn to be updated has come, it is deleted from the system; in particular, its phase is deleted from the update sequence. We will comment further on this boundary condition in section 2.2.

Second special case. When a particle j moves off the injection site of a lane (m, δ) , say at time $s - 1 + \tau_j^{m\delta}$, the algorithm needs to do more work. In that case, in order to prepare for the injection of the next particle, $j + 1$, onto that injection site, a random time interval $T_{j+1}^{m\delta}$ is drawn from the exponential distribution

$$P(T) = a e^{-aT}, \quad T > 0, \quad (2.1)$$

and $j + 1$ is scheduled to be injected at time t_{j+1} given by⁴

$$t_{j+1} = s - 1 + \tau_j + T_{j+1}^{m\delta}. \quad (2.2)$$

This implies, first, that this new particle has a phase τ_{j+1} equal to the fractional part of its time of injection, that is,

$$\tau_{j+1} = (\tau_j + T_{j+1}^{m\delta}) \bmod 1; \quad (2.3)$$

and, secondly, that its injection will occur during time step $s_{j+1} = \lfloor t_{j+1} \rfloor + 1 = s + \lfloor \tau_j + T_{j+1}^{m\delta} \rfloor$, where $\lfloor x \rfloor$ denotes the largest integer contained in x . Particle $j + 1$, its phase τ_{j+1} , its time step of injection s_{j+1} , and its lane index (m, δ) , are placed on a waiting list to which the algorithm will return during the injection step. This completes the discussion of the second special case.

Remark (i). We have $s_{j+1} \geq s$, that is, the injection may occur either during the same time step s in which the injection site was emptied, or during any later time step.

Remark (ii). The injection rule implements independent and uniformly distributed arrival times subject to interparticle hard core exclusion [25, 20].

2.1.2 Injection step

The waiting lists of each of the $2M$ lanes (m, δ) are consulted independently. If the particle, say k , on that list has a time step of injection s_k equal to the current time step s , then the algorithm implements the injection; the phase $\tau_k^{m\delta}$ is inserted in the update sequence; and the injection is considered to

⁴When no confusion can arise we will suppress the superscript ‘ $m\delta$ ’ on $\tau_j^{m\delta}$. Similarly, we will not append this superscript on quantities like t_{j+1} and k_{j+1} introduced in this subsection.

have occurred physically at time $s - 1 + \tau_k^{m\delta}$. In all following time steps, and until particle k has left the system, its position will be updated as part of the bulk step.

Remark. In the absence of any obstacles to the particle flow the above injection procedure brings the incoming lane segment into a ‘free flow configuration’, that is, one in which each particle advances by one lattice unit each time step⁵. This configuration has an average particle density $\rho^{\text{free}}(\alpha)$ and an average current $J^{\text{free}}(\alpha) = v\rho^{\text{free}}(\alpha)$ given by [25, 20]

$$\rho^{\text{free}}(\alpha) = \frac{a}{1+a}, \quad J^{\text{free}}(\alpha) = \frac{a}{1+a}, \quad (2.4)$$

where we used that $v = 1$.

2.1.3 Initial state

At time $t = 0$ we initialize the system in a random free flow configuration in which the flow has just reached the entrance sites (shown shaded in figure 1) of the intersection square, but has not entered the square itself yet. Such a configuration is conveniently generated as follows.

(1) Set time equal to $t = -L + 1$.

(2) On each of the injection sites of the $2M$ lanes independently, deposit a particle with the probability $\rho^{\text{free}}(\alpha)$ given by (2.4) and leave it unoccupied with the complementary probability $1 - \rho^{\text{free}}(\alpha)$. If a particle is deposited, assign to it a phase $\tau_0^{m\delta}$ drawn randomly and uniformly from $(0, 1)$. If no particle is deposited, draw a $T_1^{m\delta}$ from (2.1) and prepare to occupy the injection site at a time $t = -L + 1 + T_1^{m\delta}$ (that is, during some time step $s \geq -L + 2$) by a particle of phase $\tau_1^{m\delta} = T_1^{m\delta} \bmod 1$.

(3) Execute $L - 1$ time steps to take the system from time $t = -L + 1$ to $t = 0$. During these time steps the $2M$ lanes do not interact and no blocking occurs. As a result, each particle initially deposited on an injection site will at $t = 0$ occupy the corresponding entrance site, and the lane segments leading up to the intersection square will carry a free flow configuration.

Remark. Both the random initial state and the time evolution of the system are fully determined by the sequences of time intervals

$$T_1^{m\delta}, T_2^{m\delta}, T_3^{m\delta}, \dots, \quad m = 1, 2, \dots, M, \quad \delta = 'x' \text{ or } 'y', \quad (2.5)$$

which, in particular, determine the $\tau_j^{m\delta}$ for $j = 1, 2, 3 \dots$ through (2.3).

⁵A ‘free flow configuration’ is a microstate. Note that the macroscopic ‘free flow phase’ was defined in section 1 by the equality of inflow and outflow.

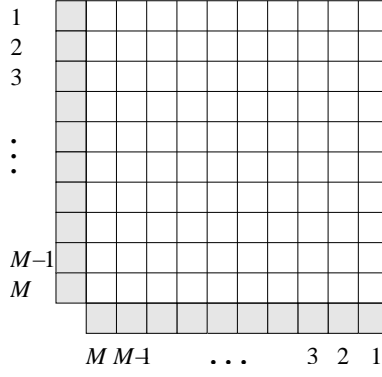


Figure 2: The $M \times M$ intersection square on which the ‘reduced’ algorithm is defined. An extra row and column of ‘entrance sites’ (shaded) along the left and lower border serve to take care of the boundary conditions.

2.2 The exit boundary

The lane segments beyond the intersection square are initially empty. The first particle to enter onto the first site of such a lane segment will from that moment on proceed unblocked at speed $v = 1$. Each next particle will enter the same segment only after its predecessor has left the first site and from then on similarly proceeds at speed $v = 1$. Hence each outgoing lane segment carries a free flow configuration and does not exert any feedback on the intersection square. The boundary condition that we applied in section 2.1.1 – namely to eliminate a particle which when its update time has come no longer has a target site – therefore correctly represents an outgoing free flow heading to infinity. We will henceforth apply this boundary condition at the immediate upper and right border of the intersection square.

3 The intersection square

We construct in this section an algorithm for simulating the crossing street model which is an exact consequence of the above ‘full’ algorithm but which is much more efficient. This ‘reduced’ algorithm retains only the particle motion on the intersection square. As we have seen, the particles exiting this square cease to play a role. We will show below that it is possible – in the limit $L \rightarrow \infty$ (see figure 1), which is precisely the one of interest – also to eliminate the incoming flows from the simulation. This is achieved with the aid of appropriate boundary conditions defined on the entrance sites shown shaded in figure 1. Hence the system geometry becomes that of figure 2. For $L = \infty$ all finite size effects are eliminated and the system has sharply defined critical points. Any remaining uncertainty on the location of these

points will be due only to the finite duration of the simulation.

3.1 Memory variables $I^{m\delta}(s)$

All statistical properties of the free flow injected into a lane (m, δ) are known. If this flow did not encounter any obstacles, then it would arrive unmodified at the entrance site L time steps later, and the boundary condition at that site would be known in a simple way. However, blocking of particles in the intersection square will typically create at the entrance site a waiting line of variable length which complicates the boundary conditions. This waiting line may be either ever growing or, instead, only fluctuating with a finite localization length, two possibilities that correspond to the lane being in the ‘jammed flow’ and the ‘free flow’ phase, respectively. The discussion of the present section applies to both cases.

The solution to the boundary condition problem will be shown to consist in introducing an auxiliary time dependent *memory variable* $I^{m\delta}(s)$. We will be able by means of this variable to determine the arrival time $t = t_j$ of each particle j at the entrance site *exactly as if*, starting at time $t = -\infty$, it had made its way through a street segment of length $L = \infty$, taking correctly into account any time it may have spent blocked in the waiting line.

We associate with each entrance site a ‘reference particle’: this will be the particle located on that site if there is one, and the first particle going to arrive on that site otherwise. Consider now the incoming segment of lane (m, δ) . During each time step every particle either advances by one lattice unit or incurs a ‘unit time delay’. We define the memory variable $I^{m\delta}(s)$ as the accumulated time delay that the reference particle in lane (m, δ) has incurred in the course of its history up to and including the s th time step. It should be noted that $I^{m\delta}(s)$ is not associated with a fixed particle; when a particle leaves the entrance site, the next one in the same lane takes over its role as the reference particle and continues to carry the variable $I^{m\delta}(s)$, the value of which is in general affected by this takeover process. The $I^{m\delta}(s)$ are nonnegative integers; for the initial state described in subsection 2.1.3 we have $I^{m\delta}(0) = 0$.

In order to determine the time evolution of the $I^{m\delta}(s)$ we examine in detail how the s th time step relates $I^{m\delta}(s)$ to $I^{m\delta}(s-1)$. In any time step three different events may occur at the entrance site of lane (m, δ) ; we denote them by the symbols ‘B’ (for ‘blocked’), ‘A’ (for ‘advancing’), and ‘E’ (for ‘empty’).

Event B. The reference particle occupies the entrance site but is blocked and does not move. In that case its delay $I^{m\delta}(s)$ is augmented by one unit,

$$I^{m\delta}(s) = I^{m\delta}(s-1) + 1. \quad (3.1a)$$

Event A. The reference particle, say j , occupies the entrance site and during the s th time step advances into the intersection square. The next particle, $j + 1$, becomes the reference particle and $I^{m\delta}(s)$, now redefined as the accumulated time delay incurred by $j + 1$, must be recalculated. If j would never have been blocked, its successor $j + 1$ would have performed the same jumps as j but with a time delay $T_{j+1}^{m\delta} + 1$. Here $T_{j+1}^{m\delta}$ is the difference between the departure time of j and the arrival time of $j + 1$ on an arbitrary site, and the extra $+1$ is the sojourn time of an unblocked particle on that site.

The extra headway $T_{j+1}^{m\delta}$ allows j to be blocked a total number of $\lfloor T_{j+1}^{m\delta} \rfloor$ times without blocking $j + 1$. Hence if the time delay $I^{m\delta}(s - 1)$ of j is small enough to satisfy $I^{m\delta}(s - 1) \leq \lfloor T_{j+1}^{m\delta} \rfloor$, then $I^{m\delta}(s) = 0$. Each supplementary blocking of j leads to a blocking, and hence to a unit time delay, of $j + 1$, and so if $I^{m\delta}(s - 1) \geq \lfloor T_{j+1}^{m\delta} \rfloor + 1$, then $I^{m\delta}(s) = I^{m\delta}(s - 1) - \lfloor T_{j+1}^{m\delta} \rfloor$. This may be combined into

$$I^{m\delta}(s) = \max(I^{m\delta}(s - 1) - \lfloor T_{j+1}^{m\delta} \rfloor, 0). \quad (3.1b)$$

We note that in event A particle $j + 1$ may or may not arrive on the entrance site during the time step s under consideration.

Event E. The reference particle does not occupy the entrance site at the beginning of the s th time step. Being free to move, it comes one lattice site closer to the entrance site during that time step. Therefore its incurred time delay remains unchanged, that is,

$$I^{m\delta}(s) = I^{m\delta}(s - 1). \quad (3.1c)$$

Again, in this event the reference particle may or may not arrive on the entrance site during time step s .

Equations (3.1a)-(3.1c) govern the time evolution of $I^{m\delta}(s)$. We are led to the important conclusion that they involve exclusively the local motion of a single reference particle on or near the entrance site of lane (m, δ) ; there is therefore no need for simulating the half-way infinite lane segment leading up to that site. Allowing the integer $I^{m\delta}$, if needed, to increase without bound, as we will do in the simulation, amounts to setting effectively $L = \infty$. The reduced algorithm of the next subsection is based on these considerations.

3.2 Reduced algorithm

The reduced algorithm is an exact consequence of the full algorithm of section 2.1.3 when the limit $L \rightarrow \infty$ is taken. This limit, almost paradoxically, simplifies the mathematics to the point that the reduced algorithm involves only the particle positions in the $M \times M$ intersection square and those in the row and column of entrance sites. There appear memory variables $I^{m\delta}$

associated with these entrance sites. As the original injection sites have moved to minus infinity, particle injection now takes place *de facto* on the entrance sites. We state below only the points of difference between the reduced algorithm and the full one described in section 2.1.3.

3.2.1 Bulk step

At the beginning of the s th time step, which covers the time interval $s - 1 \leq t < s$, the memory variables $I^{m\delta}(s - 1)$ are known. The bulk step consists again of a sweep through all particles, ordered according to increasing phases, but now all located either on the $M \times M$ square or on one of the entrance sites. Particles move as in the full algorithm of subsection 2.1.1; the ‘general case’ and the ‘first special case’ of that section are the same here.

Special case. The only special case to be discussed concerns the entrance sites. If during the s th time step the sweep encounters a particle j that is blocked on an entrance site (m, δ) , then $I^{m\delta}(s)$ is calculated from $I^{m\delta}(s - 1)$ according to (3.1a).

If during the s th time step, say at time $s - 1 + \tau_j$, the sweep encounters a particle j that advances from the entrance site into the intersection square, then a random $T_{j+1}^{m\delta}$ is drawn as in the full algorithm and $I^{m\delta}(s)$ is determined according to (3.1b). The difference $I^{m\delta}(s - 1) - I^{m\delta}(s)$ represents the reduction – compared to the free flow situation – of the number of time steps separating the departure of j and the arrival of $j + 1$ on the entrance site. Hence, recalling that $s - 1 + \tau_j$ is the time at which j leaves the entrance site, we find that $j + 1$ must be injected onto that site at time t_{j+1} given by

$$\begin{aligned} t_{j+1} &= s - 1 + \tau_j + T_{j+1}^{m\delta} + I^{m\delta}(s) - I^{m\delta}(s - 1) \\ &= s - 1 + \tau_j + T_{j+1}^{m\delta} - \min(I^{m\delta}(s - 1), \lfloor T_{j+1}^{m\delta} \rfloor), \end{aligned} \quad (3.2)$$

where to pass to the second line we used (3.1b). Equation (3.2) may be compared to (2.2). The time step of injection is $s_{j+1} = \lfloor t_{j+1} \rfloor + 1 = s + \lfloor \tau_j + T_{j+1}^{m\delta} \rfloor - \min(I^{m\delta}(s - 1), \lfloor T_{j+1}^{m\delta} \rfloor)$. Particle $j + 1$, its phase τ_{j+1} , its time step of injection s_{j+1} , and its lane index (m, δ) are placed on a list of particles waiting to be injected.

Finally, an entrance sites that is empty is not involved in the sweep. For the corresponding lane (m, δ) the memory variable remains unchanged, that is, $I^{m\delta}(s) = I^{m\delta}(s - 1)$, in agreement with (3.1c).

3.2.2 Injection step

Particle injection now takes place on the $2M$ entrance sites, to each of which the injection procedure is applied independently. If for a given entrance site (m, δ) the index s of the current time step is equal to the time step index

of the next particle to be injected on that site, say k , then the injection is carried out; the phase τ_k is inserted in the update sequence; and the injection is considered to have occurred at time $s - 1 + \tau_k$.

3.2.3 Initial state

The initial state is generated as for the full model, section 2.1.3, but with L replaced by 1, that is, the injection sites coincide with the entrance sites and step (3) of section 2.1.3 is empty. We set $I^{m\delta}(0) = 0$ for all (m, δ) .

4 Jamming transitions

Whereas the preceding sections have dealt with the microscopic algorithm, the present section is of a theoretical nature. We derive certain relations that connect the averages directly obtained in the simulation to other physically meaningful averages. As observed in the remark that closes section 2.1.3, both the initial state and the dynamics are determined by the $2M$ sequences of independent interval variables $T_j^{m\delta}$. An average is therefore a mean value calculated or measured with respect to the set $\{T_j^{m\delta}\}$. In a state with stationary currents⁶ an average over a sufficiently long period of time must be expected to coincide with this $T_j^{m\delta}$ average. Of primary interest will be the average currents $J_m^M(\alpha)$ for lane index $m = 1, 2, \dots, M$ and as a function of the injection probability α .

4.1 Current and reflection coefficient

For each lane index m we must envisage two possibilities whose actual occurrence is to be ascertained by the simulation.

(a) The incoming flow is weak enough so as to pass entirely, which means that $J_m^M(\alpha) = J^{\text{free}}(\alpha)$. The lane is then in a free flow phase.

(b) The intersection square cannot handle the full incoming flow, which means that $J_m^M(\alpha) < J^{\text{free}}(\alpha)$. The lane is then in a jammed flow phase.

We will write for either case

$$J_m^M = (1 - R_m^M) J^{\text{free}}, \quad (4.1)$$

where $R_m^M(\alpha)$ is the *reflection coefficient* of the m th lane and has the properties (1.3). The product $R_m^M J^{\text{free}}$ may be interpreted as the reflected current in the m th lane, to be discussed in greater detail in section 4.3. Since $J^{\text{free}}(\alpha)$, given by (2.4), is imposed and the $J_m^M(\alpha)$ may be measured by counting the

⁶To which we will refer as the ‘stationary state’, even though the memory variables, and concomitantly the waiting line, may or may not be stationary.

outgoing particles in each lane, substitution of these two quantities in (4.1) yields the R_m^M .

There is, however, a direct way of determining the R_m^M in the simulation. We will show below that in the long time limit the stochastic memory variable $I^{m\delta}(s)$ satisfies⁷

$$\lim_{t \rightarrow \infty} \frac{I^{m\delta}(t)}{t} = R_m^M, \quad m = 1, 2, \dots, M, \quad (4.2)$$

so that R_m^M is also the *average rate of growth* of this variable.

4.2 Equivalence of (4.1) and (4.2)

We will take (4.2) as the definition of the R_m^M and show that (4.1) follows. In the case of free flow the waiting line fluctuates only within a finite localization length, hence $I^{m\delta}(s)$ remains effectively bounded whence $R_m^M = 0$, and (4.1) is trivially true. It suffices therefore to consider the case of a jammed phase, for which $I^{m\delta}(t)$ is asymptotically linear in t . There are three steps to the proof.

First step. In every time step s one of the three equations (3.1) is applied to obtain $I^{m\delta}(s)$ from $I^{m\delta}(s-1)$. In the stationary state, let f_B , f_A , and f_E be the fractions⁸ of all time steps in which equations (3.1a), (3.1b), and (3.1c), respectively, are applied. When $I^{m\delta}(t)$ grows without bounds, the maximum in equation (3.1b) is equal to $I^{m\delta}(s-1) - \lfloor T_{j+1}^{m\delta} \rfloor$. It follows that for large times t we have the asymptotic proportionality

$$\begin{aligned} I^{m\delta}(t) &\simeq (0 \times f_E + 1 \times f_B - \kappa \times f_A) t, \\ &= (f_B - \kappa f_A) t, \end{aligned} \quad (4.3)$$

in which κ is the average of $\lfloor T_{j+1}^{m\delta} \rfloor$ and is easily calculated as

$$\kappa = \int_0^\infty dT \lfloor T \rfloor P(T) = \frac{1 - \alpha}{\alpha} \quad (4.4)$$

with $P(T)$ given by (2.1). Using (4.2), (4.3), and (4.4) we deduce that

$$R_m^M = f_B - \frac{1 - \alpha}{\alpha} f_A, \quad (4.5)$$

which completes the first step.

Second step. Using results of earlier work [25] we now determine the event fractions f_A and f_B in terms of α and β_m^M . We consider a particle on the

⁷Almost surely, in the mathematical sense.

⁸The notation leaves the dependence of these fractions on m and M implicit.

entrance site. Following [25] we let a parameter $1/\beta_m^M$ stand for the average number of time steps that such a particle has to wait before it can enter the intersection square⁹. A particle that leaves the entrance site during the s th time step may or may not be replaced during the same time step. It was shown in reference [25] that if, in the jammed phase, this site is unoccupied at some integer time s , then it will certainly be reoccupied at time $s + 1$. Let ν be the average number of particles that cross the entrance site¹⁰ between two successive integer instants of time at which this site is unoccupied. This number ν is determined exclusively by α and given by [25]

$$\frac{1}{\nu} = 1 + \frac{1}{a} - \frac{1}{\alpha}. \quad (4.6)$$

It follows that for each time step at which the entrance site is unoccupied (event E above), there are on average ν/β_m^M time steps at which it is occupied (events A and B above). Out of the latter, there are ν time steps at which the particle advances (event A), and a remaining $(1/\beta_m^M - 1)\nu$ time steps at which it stays blocked (event B). Hence when a lane is in the jammed state, the event fractions f_E , f_A , and f_B are given by

$$f_E = \frac{1}{\nu/\beta_m^M + 1}, \quad f_A = \frac{\nu}{\nu/\beta_m^M + 1}, \quad f_B = \frac{(1/\beta_m^M - 1)\nu}{\nu/\beta_m^M + 1}. \quad (4.7)$$

Substituting (4.7) in (4.5) we obtain

$$R_m^M = \frac{\nu\beta_m^M}{\nu + \beta_m^M} \left(\frac{1}{\beta_m^M} - \frac{1}{\alpha} \right). \quad (4.8)$$

Only for positive R_m^M is this result consistent with our initial supposition that lane m is in the jammed phase. Therefore, (4.8) shows that lane m is jammed when $\alpha > \beta_m^M$. Remembering that β_m^M is a function, although unknown, of α and M we see that the jamming point $\alpha = \alpha_m^M$ is the solution of $\beta_m^M(\alpha) = \alpha$.

Third step. In the jammed phase the particle density on any site of the waiting line, and in particular on the entrance site, is given by $\rho^{\text{jam}} = f_A + f_B$ and the current entering the intersection square is equal to $\beta_m^M \rho^{\text{jam}}$. Since in the stationary state, the current entering the intersection square in lane m is equal to the outgoing current J_m^M in that lane, it follows with the aid of

⁹The inverse β_m^M is the probability per time step that a particle on the entrance site is allowed to enter the intersection square. In reference [25] this parameter (called β there) was an independent control parameter and entrance events of successive particles were uncorrelated. In the present case β_m^M is a complicated function of α and M ; moreover, correlations must be expected between entrance events.

¹⁰This sequence of particles is called a ‘platoon’.

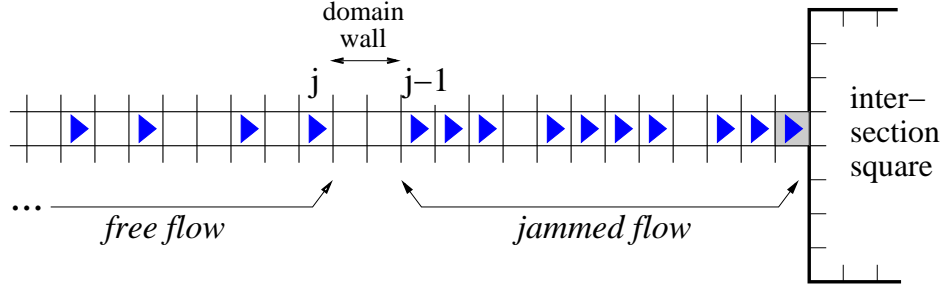


Figure 3: Lane containing a domain wall that separates a free flow from a jammed flow region. The particles in the latter constitute the waiting line. The particle on the entrance site (shaded) is the reference particle for the lane's memory variable.

(4.7) that

$$\rho^{\text{jam}}(\alpha, \beta_m^M) = \frac{\nu}{\nu + \beta_m^M}, \quad J_m^M(\alpha, \beta_m^M) = \frac{\nu \beta_m^M}{\nu + \beta_m^M}, \quad (4.9)$$

which is the jammed phase counterpart to (2.4). Upon eliminating β_m^M from (4.8) and (4.9), solving for J_m^M in terms of R_m^M , and using (2.4), we obtain (4.1). We have therefore established the interpretation of R_m^M , initially defined by (4.2), as the reflection coefficient of the m th lane. This completes the equivalence proof of this subsection.

Equations (4.1), (4.8), and (4.9) show that out of the three quantities R_m^M , J_m^M , and β_m^M , each one determines the two others.

4.3 Reflected current and domain wall motion

We consider a single lane m and suppose that $\alpha > \alpha_m^M$ so that jamming will occur. We now refer to figure 3. Going from left to right, the first particle in the waiting line is by convention the leftmost particle ever to have been blocked; in the figure this is particle $j - 1$ ¹¹. We will say that a ‘domain wall’ is located immediately to the left of this particle. The waiting line constitutes a finite spatial domain of jammed flow, separated by the domain wall from a half-infinite region in the free flow phase. The domain wall *motion* in a finite one-dimensional system has been discussed in references [25, 28].

Our reduced algorithm no longer contains the description of the domain walls. We will show below, however, that if at time s we know $I^{m\delta}(s)$ for some lane (m, δ) , we can relate this quantity analytically to the *average* position that the domain wall in that lane would have at that time. For the initial

¹¹This cannot be concluded from the figure alone, since the property of having never been blocked depends on the history of the configuration.

conditions of section 3.2.3 the domain wall is located at the entrance of the intersection square and, for $\alpha > \alpha_m^M$, starts at time $t = 0$ propagating in the negative direction at some average speed that we will call¹² \mathbf{v}_m^M . If we assimilate the incoming particle flow to a wave, then the domain wall is the moving front of the reflected wave.

Consider now a time T large enough for fluctuating variables to be approximated by their averages. Let \mathbf{L}_m^M be the linear size of the jammed flow region, so that $\mathbf{v}_m^M = \mathbf{L}_m^M/T$. To determine \mathbf{L}_m^M we reason as follows. The average number of particles \mathbf{N}_m^M that up to time T has been prevented to cross the interaction square is

$$\mathbf{N}_m^M = T(J^{\text{free}} - J_m^M). \quad (4.10)$$

These particles are spread out along the waiting line and therefore the density ρ^{jam} of this line is equal to ρ^{free} plus the extra contribution $\mathbf{N}_m^M/\mathbf{L}_m^M$. This leads to a continuity equation in the form

$$\rho^{\text{jam}}(\alpha, \beta_m^M) = \rho^{\text{free}}(\alpha) + \frac{T}{\mathbf{L}_m^M} [J^{\text{free}}(\alpha) - J_m^M(\alpha, \beta_m^M)], \quad (4.11)$$

in which $\rho^{\text{jam}}(\alpha, \beta_m^M)$ and $\rho^{\text{free}}(\alpha)$ are known from (4.9) and (2.4), respectively. Using that $\mathbf{v}_m^M = \mathbf{L}_m^M/T$ we deduce that the speed of propagation \mathbf{v}_m^M of the domain wall is given by

$$\begin{aligned} \mathbf{v}_m^M &= \frac{\rho^{\text{free}} - \beta_m^M \rho^{\text{jam}}}{\rho^{\text{jam}} - \rho^{\text{free}}} \\ &= \left(1 - \frac{\beta_m^M}{\alpha}\right) \left(\frac{1 - \beta_m^M}{\nu} + \frac{1}{\alpha} - 1\right)^{-1}. \end{aligned} \quad (4.12)$$

The second line results from some algebra in which (4.9) and (2.4) are used; the expression may be rewritten in several other ways. When eliminating β_m^M from (4.12) and (4.8) we obtain the speed of propagation \mathbf{v}_m^M of the reflected wave as a function of \mathbf{R}_m^M . Explicitly,

$$\mathbf{v}_m^M = \frac{\alpha \nu \mathbf{R}_m^M}{\alpha \mathbf{R}_m^M + (1 - \alpha) \nu}, \quad (4.13)$$

in which the coefficients are known functions of only the particle injection rate α . This equation shows that we may retrieve the average domain wall position from the reduced algorithm, in spite of the fact that at the microscopic level the waiting line has been eliminated from the description. Equation (4.13) is valid only for $\alpha \geq \alpha_m^M$. When α decreases to α_m^M , we have that $\mathbf{R}_m^M \rightarrow 0$ and hence $\mathbf{v}_m^M \simeq [\alpha/(1 - \alpha)] \mathbf{R}_m^M \rightarrow 0$.

¹²Quantities related to this wall will be indicated by sans serif symbols.

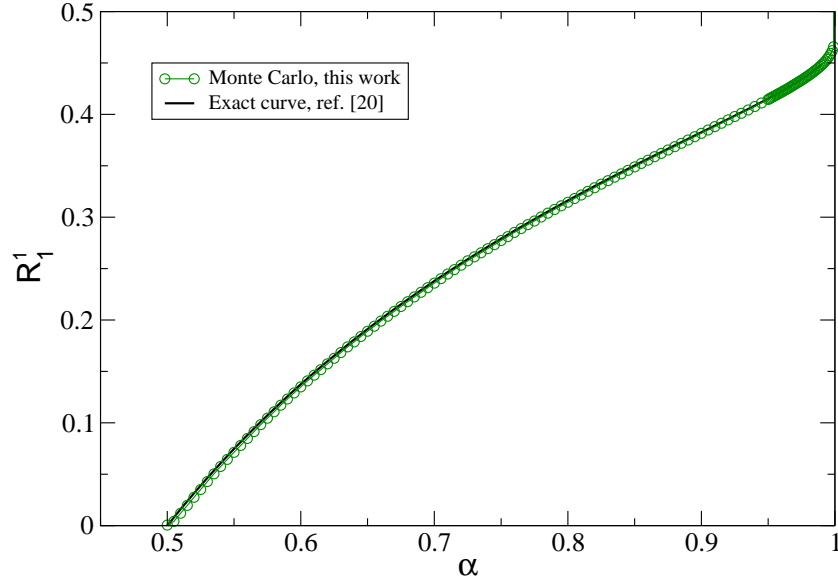


Figure 4: Reflection coefficient $R_1^1(\alpha)$ for $M = 1$. The Monte Carlo algorithm of this work, which simulates an infinite system, agrees very well with the known analytic result [20].

5 Monte Carlo results

Using the reduced algorithm of section 3 we have carried out simulations of the intersecting streets for values of M up to $M = 24$. We determined the phase diagram of the stationary state of the intersection square as a function of the injection probability α and the linear size M . In our figures we will present the reflection coefficients $R_m^M(\alpha)$, which by (1.2) are directly equivalent to the currents $J_m^M(\alpha)$. Each simulation was started at $t = 0$ from the initial random free flow configuration described in section 3.2.3. Relaxation to a stationary state appeared to be very rapid.

It is important to stress that, since we are in the limit $L = \infty$, the simulation is *free of finite size effects*. The remaining statistical errors in the simulation results are entirely due to the finiteness of the simulation time.

5.1 Phase diagram for $M = 1$ and $M = 10$

For $M = 1$ the intersection square consists of a single site and there is a single critical jamming point $\alpha_1^1 = \frac{1}{2}$ as predicted in earlier work [20]. The $M = 1$ simulation involves less than a dozen variables, in spite of the fact that we simulate an infinite system. Figure 4 shows the simulation results for the reflection coefficient $R_1^1(\alpha)$. Each data point results from an average over 1.1×10^6 time steps and over the x and y direction. One observes that R_1^1 is

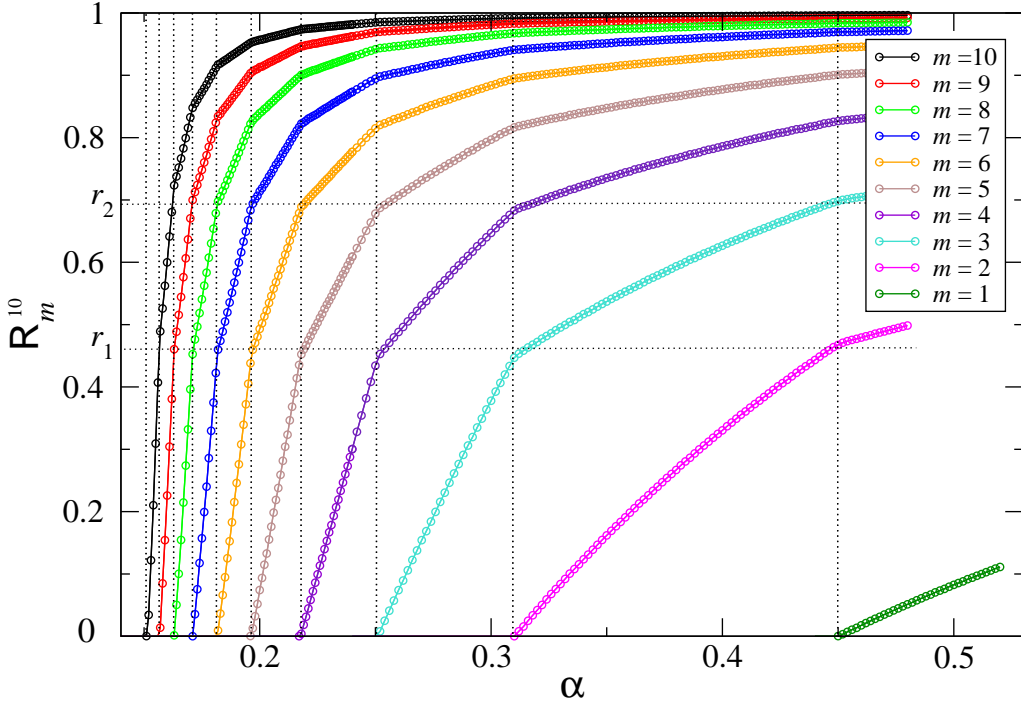


Figure 5: Reflection coefficient $R_m^{10}(\alpha)$ for $M = 10$ and $m = 1, 2, \dots, 10$. The lines connecting the points are meant to guide the eye. The thin dotted vertical lines correspond to the critical points on the α axis; in increasing order these are $\alpha_{10}^{10}, \alpha_9^{10}, \dots, \alpha_1^{10}$. The thin dotted horizontal lines are the approximate levels at which the curves have their first and second discontinuity of slope.

nonzero only for $\alpha > \frac{1}{2}$ and that it tends to $\frac{1}{2}$ with infinite slope when $\alpha \rightarrow 1$. A different representation of figure 4 was obtained analytically¹³ by Appert-Rolland *et al.* [20], whose results $J^{\text{free}} = a/(1+a)$ and $J_1^1 = \nu/(2\nu+1)$ imply that $R_1^1 = (\nu - \nu/a + 1)/(2\nu + 1)$. This theoretical curve is also shown in the figure; its full agreement with the Monte Carlo data confirms the correctness of the ‘reduced’ algorithm of this paper.

Figure 5 is for $M = 10$ and presents the reflection coefficients $R_m^{10}(\alpha)$ for $m = 1, 2, \dots, 10$. Each data point results from an average over 1.1×10^7 time steps and the statistical error is less than the symbol size. There are ten critical values $\alpha_{10}^{10} < \alpha_9^{10} < \dots < \alpha_1^{10}$. For $\alpha_{m+1}^{10} < \alpha < \alpha_m^{10}$ the outer lanes $1, 2, \dots, m$ are in a free flow phase and the inner ones $m+1, \dots, M-1, M$ have jammed flow. For each m the data shown are averages on the two lanes (m, x) and (m, y) . We verified that no spontaneous symmetry breaking

¹³The $M = 1$ system actually studied in [20] was more general: it had four control parameters (two entrance and two exit rates) which could break the symmetry between the two crossing lanes; in the present work the two streets are symmetric.

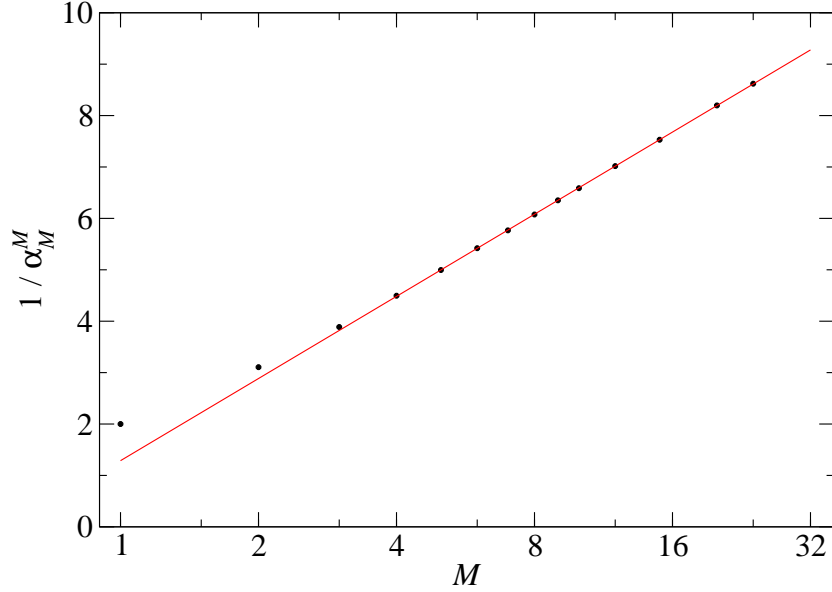


Figure 6: Inverse of the principal critical point α_M^M as a function of the linear size M of the intersection square (notice the logarithmic scale of M). The errors in the data points are smaller than the symbol sizes. The straight line is the best asymptotic linear fit.

occurs between the x and the y direction.

5.2 Phase diagram for general M

We performed simulations for $M = 1, 2, \dots, 10, 12, 15, 20, 24$. Our findings for the other M values are qualitatively similar to those obtained in figure 5 for $M = 10$. Again, there is no spontaneous symmetry breaking between the two perpendicular directions. As α increases, the system first reaches its principal critical point α_M^M at which the two innermost lanes (with $m = M$) get jammed. Then there is an initially fast succession of critical points $\alpha_M^M < \alpha_{M-1}^M < \alpha_{M-2}^M < \dots$ at each of which a further pair of lanes, one in each street, gets jammed. The spacing between the critical points becomes gradually larger and at the last critical point, $\alpha = \alpha_1^M$, the outermost pair of lanes (with $m = 1$) gets jammed.

The principal critical point α_M^M decreases with M but its exact law is difficult to ascertain. It is very well approximated by

$$\alpha_M^M \simeq \frac{1}{A + B \log M}, \quad M \gtrsim 4, \quad (5.1)$$

as shown in figure 6, where $A = 1.287$ and $B = 2.306$. The high precision of these results is due to the fact that even if the street *widths* M are finite, the

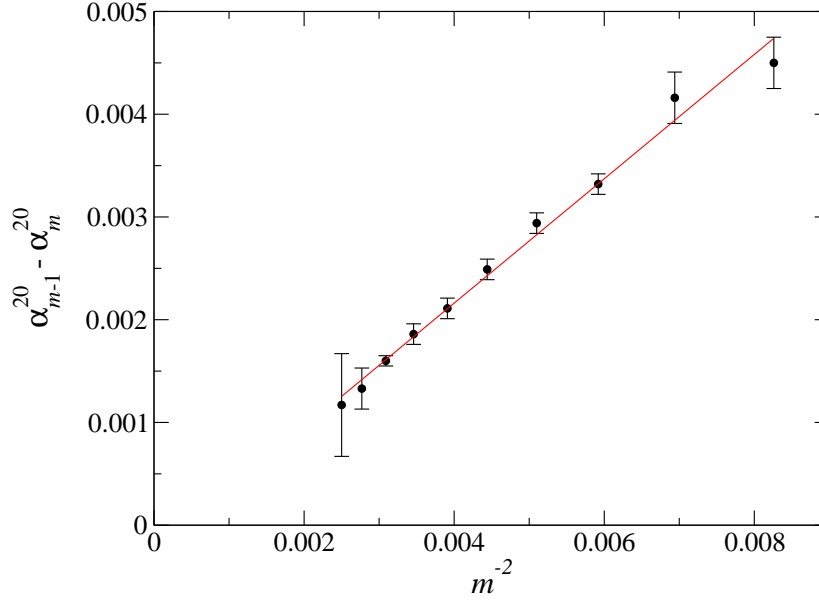


Figure 7: The intervals $\alpha_{m-1}^M - \alpha_m^M$ as a function of m^{-2} for $M = 20$. The straight line is the best linear fit.

simulations correspond to infinite street *length*, $L = \infty$, and hence there are no finite length effects. The question of the large M limit of the critical point was asked also in the context of the BHL model [21, 34]. All these authors have declared to be unable to state whether or not this point goes to zero in the limit $M \rightarrow \infty$. On the basis of our above results, albeit for relatively small M values, one might guess that (5.1) is the correct asymptotic law and hence that for the model of this work the principal critical point does tend to zero with increasing M . We refrain at present, however, from drawing this conclusion. The reason is that preliminary simulations for larger M reveal complications in the form of metastabilities associated with the transition possibly turning first order. Since the infinite system limit is a fundamental question in statistical mechanics, we believe it is worthwhile, and necessary, to spend further efforts on studying it carefully.

The intervals between two successive critical points scale in fairly good approximation as

$$\alpha_{m-1}^M - \alpha_m^M \simeq C_M + \frac{D_M}{m^2}, \quad (5.2)$$

where C_M is a slightly negative and D_M a positive constant. Equation (5.2) appears to hold for m not too small and up to $m = M$. For $M = 20$ this is shown in figure 7; the fit has $C_{20} = -0.00026$ and $D_{20} = 0.605$.

For $\alpha > \alpha_m^M$ the curve $R_m^M(\alpha)$ is composed of m segments that join with a discontinuity of slope at the higher critical values $\alpha_{m-1}^M, \alpha_{m-2}^M, \dots, \alpha_1^M$. For

M and m not too small, these segments, although actually curved, are very close to linear. The slope of the first segment of $R_m^M(\alpha)$, evaluated at α_m^M , is approximately proportional to m^2 . Therefore, the larger M , the steeper the initial rise, and the finer should be the grid of points on the α axis in order to obtain a resolution of the successive jamming transitions in the inner lanes. This renders the determination of $R_m^M(\alpha)$ by simulation increasingly harder as M grows.

The first discontinuity of slope in each of the $R_m^M(\alpha)$ appears when $R_m^M(\alpha)$ reaches a level r_1 that is, again in fairly good approximation, m independent. This had to be expected from the scaling observed above for the first segment of $R_m^M(\alpha)$, whose horizontal and vertical extension are $\sim m^{-2}$ and $\sim m^2$, respectively. For $M = 10$ figure 5 shows that this level value is $r_1 \approx 0.47$. Similarly, the second discontinuity appears at $R_m^M(\alpha) \approx r_2 \approx 0.69$, the third one (not explicitly indicated in the figure) at $r_3 \approx 0.83$, and so on. The values of these levels r_i appear to depend only little on M , as anticipated in the notation.

As M grows, the values α_m^M with fixed m seem to tend to a limit value that may be called α_m^∞ , but that we have not tried to determine with any precision in this study. Finally, for fixed α and M , the reflection coefficient approaches unity, and hence the transmitted current vanishes, roughly exponentially with m . It is still a challenge to find an analytic explanation for all these qualitative and quantitative observations, which were not expected *a priori*. Whereas an exact solution seems beyond reach, we think that an approximate theory may be possible.

5.3 Snapshots

Figure 8 is a snapshot of a 10×10 intersection square simulated with an injection probability $\alpha = 0.169$. It is the only one of our simulations that was carried out with the ‘full’ algorithm. The length of the incoming streets is equal to $L = 15$. For this value of α figure 5 shows that the lanes $m = 1, 2, \dots, 7$ are in the free flow phase whereas those with $m = 8, 9, 10$ are in the jammed flow phase. This is roughly what is visible in figure 8; nevertheless, considerable fluctuations occur between independent snapshots, with lanes getting jammed and opening up intermittently. These fluctuations are enhanced by the fact that for finite L there are no sharply defined transition points, L being automatically an upper limit on the length of the waiting lines.

Figure 9 is a snapshot of a 20×20 intersection square together with its row and column of entrance sites. The injection probability is $\alpha = 0.15$. For $M = 20$ a phase diagram analogous to figure 5 shows that $\alpha_{11}^{20} < 0.15 < \alpha_{10}^{20}$. As a consequence, for $\alpha = 0.15$ lanes 1 through 10 are in the free flow phase and 11 through 20 are in the jammed flow phase.

Snapshots of this system taken at sufficiently long time intervals show

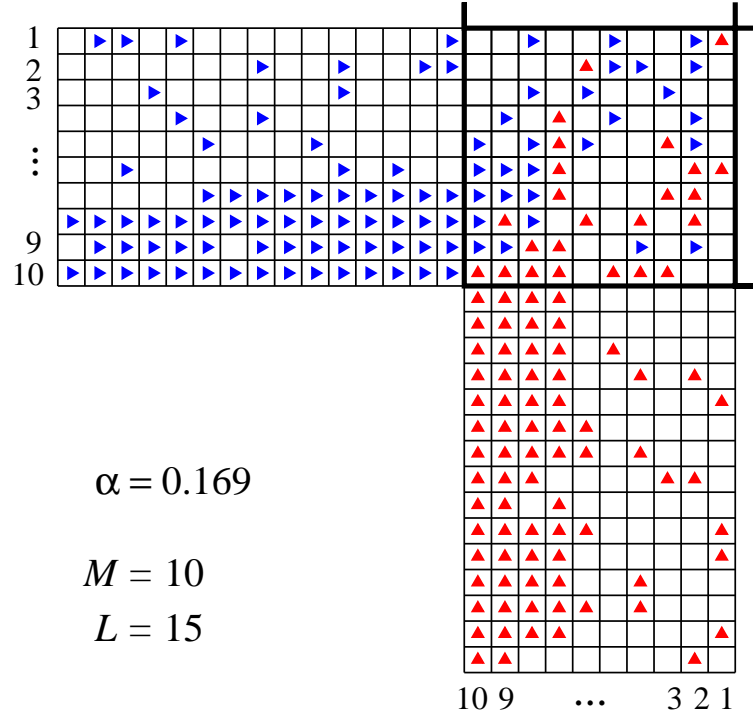


Figure 8: Snapshot of an intersection square of linear size $M = 10$ and for injection probability $\alpha = 0.169$, obtained by the ‘full’ algorithm with $L = 15$. For this value of α , lanes 1 through 7 are in the free flow and 8, 9, 10 in the jammed flow phase.

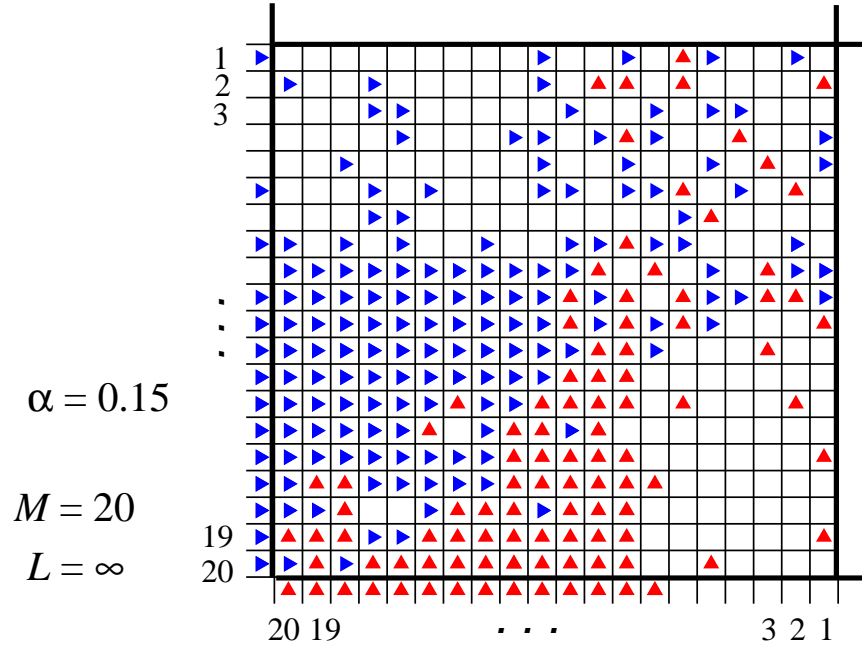


Figure 9: Snapshot of an intersection square of linear size $M = 20$ for injection probability $\alpha = 0.15$, obtained by the ‘reduced’ algorithm ($L = \infty$). The row and column of entrance sites are also shown. For this value of α lanes 1 through 10 are in the free flow and 11 through 20 in the jammed flow phase.

considerable variation, but all have in common the existence of an approximately square (or sometimes rectangular) high density region in the lower left corner of the intersection square. The shape of this corner region fluctuates, since the jammed phase current still lets particles pass relatively easily along the outermost jammed lanes (with indices $m = 11, 12, \dots$) of this region. However, the jammed phase current in the innermost lanes (those with indices $m = \dots, 18, 19, 20$) is very close to zero and the particles in those lanes are in a quasi-permanent frozen state. The net result is that there occurs a “freezing out” of a set of inner lanes, which reduces the traffic problem on the 20×20 intersection square to an effective one on a smaller square. In figure 8, this effective intersection square would correspond to a square of size 10×10 in the upper right corner. This reduction is certainly not exact but may well offer the starting point for a first theoretical approach.

5.4 Further comments

Our algorithm for crossing streets of infinite length was reduced to an algorithm on the finite-size intersection square with special memory boundary conditions. It is therefore natural that we make a comparison with existing models on finite lattices. A prominent one is the BHL model due to Biham *et al.* [21]. In this model two types of particles move unidirectionally on a torus of size $M \times M$, one type horizontally and the other one vertically. In the original version of this system the horizontally and vertically moving particles are updated in parallel at the even and odd time steps, respectively. A single phase transition was observed between a free flow phase and a fully jammed (zero flow) phase. The importance of the aspect ratio of the lattice was stressed by D’Souza [29], who also showed that there exists a third, high density, jammed phase. In reference [21] and in the work that it has sparked [30, 31, 32, 33, 29], several variants of this BHL model were studied, many of them introducing additional stochastic elements. Of particular interest in our context is a recent brief report by Ding *et al.* [34]. These authors considered the BHL model with the *standard* open boundary conditions that consist in filling an empty site on the left or bottom boundary with a fixed probability at each time step. Under such boundary conditions there is no place for waiting lines and no sharply defined phase transition can be expected as long as M is finite. Obviously, the two types of boundary conditions correspond to distinct driving parameters and have distinct sets of applications: urban road traffic for the BHL model and intersecting pedestrian traffic flows for ours.

The model of reference [34] also differs from ours by its use of random sequential update; but although the update type is important for the interpretation of the model, we consider that difference as secondary.

The most striking similarity between the results of reference [34] and ours is that when the driving parameter increases, the $M \times M$ domain is gradually filled with a dense phase that occupies a square or rectangular lower left corner, in the way shown in our figure 9. We find, as in reference [34], that the complementary upper right corner, which is in a free flow state, has a size roughly independent of M .

As for the differences, our $M \times M$ domain is part of two infinite streets and the memory boundary conditions keep track of the waiting lines in each of the $2M$ lanes. These boundary conditions create long-range correlations in time, that in turn determine sharply defined phase transition points on the α axis for all $M = 1, 2, 3, \dots$. As a consequence, in the model of this paper the progressive growth of the dense corner occurs through a sequence of M phase transitions. Our simulations concern relatively modest values of M and are aimed at locating the transition points with high precision. In addition we are able to assign to each of the $2M$ lanes a reflection coefficient which serves as a lane order parameter, and a waiting line of which at each instant of time the average length is known.

6 Conclusion

We have introduced and studied a lattice model of pedestrian traffic on two crossing one-way streets. Each street is represented by a set of M parallel TASEP lanes and the only model parameter besides M is the injection probability α of a pedestrian ('particle') at minus infinity. The dynamics is based on frozen shuffle update [24, 25, 20]: particles enter stochastically but once in the system move deterministically. From an algorithmic point of view we have found that in this model the frozen shuffle update leads to accurate results even with a modest simulation effort. We consider this as an encouragement for simulating other models, possibly very different ones, with the same update.

The intersecting streets that we considered are infinite in both directions, but one achievement of this work has been to show that the dynamics may be reduced to a problem of interacting variables on the sites of the finite $M \times M$ intersection square. Appropriate boundary conditions were formulated in terms of 'memory variables' and we established the theoretical relation between these new quantities and the outgoing current.

Our Monte Carlo work shows that as α increases, the system undergoes a sequence of M phase transitions starting with the one at the principal critical point α_M^M . Since we are able to perform the simulation directly on an infinite system, there are no finite size effects; the uncertainty in the critical points is due only to the finiteness of the simulation time. At each transition a new pair of lanes, one in each street, passes from a free flow to a jammed flow phase.

We find that a reflection coefficient for the current is an appropriate order parameter. The accuracy of the algorithm has allowed us to establish certain surprising features, such as the discontinuities of slope that occur when the reflection coefficient crosses a set of narrowly defined level values. All these results should help trigger interest in building theoretical approaches to this and similar systems.

The present study stays far from exploring, let alone answering, all questions that this specific model poses. Quantities of interest not studied here include the particle density in the intersection square, its space dependence, and its fluctuations and correlations; the fluctuation of the waiting line lengths at and near the critical points; and its relation to the fluctuation of the memory variables. An unanswered question also concerns the behavior of the principal critical point α_M^M as $M \rightarrow \infty$.

We believe that this model, because of its simplicity, sets a standard scenario with respect to which others may be discussed. Clearly, ideas come readily to mind about how to modify or extend this model, for example by considering streets of different widths M_1 and M_2 , unequal injection probabilities α_x and α_y , and so on; or by opening the possibility for pedestrians to move laterally or diagonally forward. Sideways steps between the lanes are certainly expected to blur the distinction between the jamming transitions in the individual lanes that here occur separately. It is hard to guess, however, what sort of a transition will then result. Several effects of the present model may turn out to be robust, such as the predominance of the flow through the outer lanes over those through the inner ones. In future work [35] we will address a few of the many new issues raised here.

Acknowledgments

The authors thank L. Santen for pointing out to them several important references. They also thank J. Cividini, J.-M. Caillol, and R.K.P. Zia for their interest shown during various stages of this work.

References

- [1] Andreas Schadschneider. Modelling of transport and traffic problems. *Lecture Notes in Computer Science*, 5191:22–31, 2008.
- [2] D. Helbing. Traffic and related self-driven many-particle systems. *Reviews of Modern Physics*, 73:1067–1141, 2001.
- [3] Nicola Bellomo and Christian Dogbe. On the modeling of traffic and crowds a survey of models, speculations, and perspectives. *SIAM Review*, 53:409–463, 2011.

- [4] T. Vicsek and A. Zafeiris. Collective motion. [arXiv:1010.5017v2](#).
- [5] C. Burstedde, K. Klauck, A. Schadschneider, and J. Zittartz. Simulation of pedestrian dynamics using a 2-dimensional cellular automaton. *Physica A*, 295:507–525, 2001.
- [6] A. Schadschneider. Cellular automaton approach to pedestrian dynamics - theory. In M. Schreckenberg and S.D. Sharma (Eds.), editors, *Pedestrian and Evacuation Dynamics*, pages 75–85. Springer, 2002.
- [7] A. Kirchner, K. Nishinari, and A. Schadschneider. Friction effects and clogging in a cellular automaton model for pedestrian dynamics. *Phys. Rev. E*, 67:056122, 2003.
- [8] H. Klüpfel. The simulation of crowds at very large events. In A. Schadschneider, T. Poschel, R. Kuhne, M. Schreckenberg, and D.E. Wolf, editors, *Traffic and Granular Flow '05*, pages 341–346, 2007.
- [9] D. Chowdhury, L. Santen, and A. Schadschneider. Statistical physics of vehicular traffic and some related systems. *Physics Reports*, 329:199–329, 2000.
- [10] B. Derrida. An exactly soluble non-equilibrium system: the asymmetric simple exclusion process. *Phys. Reports*, 301:65, 1998.
- [11] J. Brankov, N. Pesheva, and N. Bunzarova. Totally asymmetric exclusion process on chains with a double-chain section in the middle: Computer simulations and a simple theory. *Phys. Rev. E*, 69:066128, 2004.
- [12] E. Pronina and A.B. Kolomeisky. Theoretical investigation of totally asymmetric exclusion processes on lattices with junctions. *J. Stat. Mech.*, P07010, 2005.
- [13] D.-W. Huang. Analytical results of asymmetric exclusion processes with ramps. *Phys. Rev. E*, 72:016102, 2005.
- [14] D.-W. Huang. Ramp induced transitions in traffic dynamics. *Phys. Rev. E*, 73:016123, 2006.
- [15] K. Nagel and M. Schreckenberg. A cellular automaton model for freeway traffic. *J. Physique I* 2:2221–2229, 1992.
- [16] M.E. Foolaadvand, Z. Sadjadi and M.R. Shaeabani. Optimised Traffic Flow at a Single Intersection: Traffic Responsive signalisation. *J. Phys. A - Math. Gen.*, 37:561, 2004.

- [17] M.E. Foulaadvand and M. Neek-Amal. Asymmetric simple exclusion processes describing conflicting traffic flows. *Europhys. Lett.*, 80:60002, 2007.
- [18] H.-F. Du, Y.-M. Yuan, M.-B. Hu, R. Wang, R. Jiang, and Q.-S. Wu. Totally asymmetric exclusion processes on two intersected lattices with open and periodic boundary conditions. *J. Stat. Mech.*, P03014, 2010.
- [19] M.E. Foulaadvand and S. Belbasi. Vehicular traffic flow at an intersection with the possibility of turning. *Preprint arXiv:1105.1445 [physics.soc.ph]*
- [20] C. Appert-Rolland, J. Cividini, and H.J. Hilhorst. Intersection of two TASEP traffic lanes with frozen shuffle update. *J. Stat. Mech. (2011) P10014*.
- [21] O. Biham, A.A. Middleton, and D. Levine. Self-organization and a dynamical transition in traffic-flow models. *Phys. Rev. A* 46:R6124, 1992.
- [22] B. Schmittmann and R.K.P. Zia. Statistical Mechanics of Driven Diffusive Systems. In *Phase Transitions and Critical Phenomena* edited by C. Domb and J.L. Lebowitz. Academic Press, New York, Vol. 17, 1995.
- [23] B. Schmittmann and R.K.P. Zia. Driven diffusive systems. An introduction and recent developments. *Physics Reports*, 301:4564, 1998
- [24] C. Appert-Rolland, J. Cividini, and H.J. Hilhorst. Frozen shuffle update for an asymmetric exclusion process on a ring. *J. Stat. Mech. (2011) P07009*.
- [25] C. Appert-Rolland, J. Cividini, and H.J. Hilhorst. Frozen shuffle update for a deterministic totally asymmetric exclusion process with open boundaries. *J. Stat. Mech. (2011) P10013*.
- [26] M. Wölki, A. Schadschneider, and M. Schreckenberg. Asymmetric exclusion processes with shuffled dynamics. *J. Phys. A-Math. Gen.*, 39:33–44, 2006.
- [27] D.A. Smith and R.E. Wilson. Dynamical pair approximation for cellular automata with shuffle update. *J. Phys. A: Math. Theor.*, 40(11):2651–2664, 2007.
- [28] L. Santen and C. Appert. The asymmetric exclusion process revisited: Fluctuations and dynamics in the domain wall picture. *J. Stat. Phys.*, 106:187–199, 2002.

- [29] R. M. D'Souza. Coexisting phases and lattice dependence of a cellular automaton for traffic flow. *Phys. Rev.E* 71:066112, 2005.
- [30] J.A. Cuesta, F.C. Martínez, J.M. Molera, and A. Sánchez. Phase transitions in two-dimensional traffic-flow models. *Phys. Rev. E* 48:R4175, 1993.
- [31] J.M. Molera, F.C. Martínez, J.A. Cuesta, and R. Brito. Theoretical approach to two-dimensional traffic flow models. *Phys. Rev. E* 51:175-187, 1995.
- [32] T. Nagatani. Jamming transition in the traffic-flow model with two-level crossings. *Phys. Rev. E* 48:3290-3294, 1993.
- [33] A. Benyoussef, H. Chakib, and H. Ez-Zahraouy. Anisotropy effect on two-dimensional cellular-automata traffic flow with periodic and open boundaries. *Phys. Rev. E* 68:026129, 2003.
- [34] Z.-J. Ding, R. Jiang, and B.-H. Wang. Traffic flow in the Biham-Middleton-Levine model with random update rule. *Phys. Rev. E* 83:047101, 2011.
- [35] J. Cividini, C. Appert-Rolland, and H.J. Hilhorst. *In preparation.*